**ADD -- *Add***

|  |  |
| --- | --- |
| Description: | Adds two registers and stores the result in a register |
| Operation: | $d = $s + $t; advance\_pc (4); |
| Syntax: | add $d, $s, $t |
| Encoding: | 0000 00ss ssst tttt dddd d000 0010 0000 |

**ADDI -- *Add immediate (with overflow)***

|  |  |
| --- | --- |
| Description: | Adds a register and a sign-extended immediate value and stores the result in a register |
| Operation: | $t = $s + imm; advance\_pc (4); |
| Syntax: | addi $t, $s, imm |
| Encoding: | 0010 00ss ssst tttt iiii iiii iiii iiii |

**BEQ -- *Branch on equal***

|  |  |
| --- | --- |
| Description: | Branches if the two registers are equal |
| Operation: | if $s == $t advance\_pc (offset << 2)); else advance\_pc (4); |
| Syntax: | beq $s, $t, offset |
| Encoding: | 0001 00ss ssst tttt iiii iiii iiii iiii |

### J -- *Jump*

|  |  |
| --- | --- |
| Description: | Jumps to the calculated address |
| Operation: | PC = nPC; nPC = (PC & 0xf0000000) | (target << 2); |
| Syntax: | j target |
| Encoding: | 0000 10ii iiii iiii iiii iiii iiii iiii |

### LUI -- *Load upper immediate*

|  |  |
| --- | --- |
| Description: | The immediate value is shifted left 16 bits and stored in the register. The lower 16 bits are zeroes. |
| Operation: | $t = (imm << 16); advance\_pc (4); |
| Syntax: | lui $t, imm |
| Encoding: | 0011 11-- ---t tttt iiii iiii iiii iiii |

### LW -- *Load word*

|  |  |
| --- | --- |
| Description: | A word is loaded into a register from the specified address. |
| Operation: | $t = MEM[$s + offset]; advance\_pc (4); |
| Syntax: | lw $t, offset($s) |
| Encoding: | 1000 11ss ssst tttt iiii iiii iiii iiii |

**ORI -- *Bitwise or immediate***

|  |  |
| --- | --- |
| Description: | Bitwise ors a register and an immediate value and stores the result in a register |
| Operation: | $t = $s | imm; advance\_pc (4); |
| Syntax: | ori $t, $s, imm |
| Encoding: | 0011 01ss ssst tttt iiii iiii iiii iiii |

**SW -- *Store word***

|  |  |
| --- | --- |
| Description: | The contents of $t is stored at the specified address. |
| Operation: | MEM[$s + offset] = $t; advance\_pc (4); |
| Syntax: | sw $t, offset($s) |
| Encoding: | 1010 11ss ssst tttt iiii iiii iiii iiii |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Register Number** | **Usage** | **Preserved by callee?** |
| $zero | 0 | hardwired 0 | N/A |
| $v0-$v1 | 2-3 | return value and expression evaluation | no |
| $a0-$a3 | 4-7 | arguments | no |
| $t0-$t7 | 8-15 | temporary values | no |
| $s0-$s7 | 16-23 | saved values | YES |
| $t8-$t9 | 24-25 | more temporary values | no |
| $gp | 28 | global pointer | YES |
| $sp | 29 | stack pointer | YES |
| $fp | 30 | frame pointer | YES |
| $ra | 31 | return address | YES |